Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Ciência da Computação

Título do trabalho

um subtítulo

Antonio Marcos Shiro Arnauts Hachisuca

Monografia Final

mac 499 — Trabalho de
Formatura Supervisionado

Supervisor: Prof. Dr. Marcel Kenji de Carli Silva

São Paulo
2024

Esta seção é opcional e fica numa página separada;
ela pode ser usada para uma dedicatória ou epígrafe.

Agradecimentos

Do. Or do not. There is no try.

— Mestre Yoda

Texto texto. Texto opcional.

Resumo

Antonio Marcos Shiro Arnauts Hachisuca. **Título do trabalho: *um subtítulo***. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2024.

Elemento obrigatório, constituído de uma sequência de frases concisas e objetivas, em forma de texto. Deve apresentar os objetivos, métodos empregados, resultados e conclusões. O resumo deve ser redigido em parágrafo único, conter no máximo 500 palavras e ser seguido dos termos representativos do conteúdo do trabalho (palavras-chave). Deve ser precedido da referência do documento. Texto texto. Texto texto.

Palavras-chave: Palavra-chave1. Palavra-chave2. Palavra-chave3.

Abstract

Antonio Marcos Shiro Arnauts Hachisuca. **Title of the document: *a subtitle***. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2024.

Elemento obrigatório, elaborado com as mesmas características do resumo em língua portuguesa. De acordo com o Regimento da Pós-Graduação da USP (Artigo 99), deve ser redigido em inglês para fins de divulgação. É uma boa ideia usar o sítio www.grammarly.com na preparação de textos em inglês. Text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text. Text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text.

Keywords: Keyword1. Keyword2. Keyword3.

Lista de abreviaturas

URL	Localizador Uniforme de Recursos (*Uniform Resource Locator*)
IME	Instituto de Matemática e Estatística
USP	Universidade de São Paulo

Lista de símbolos

Lista de figuras

Lista de tabelas

Lista de programas

Sumário

Introdução

Alguma coisa.

Capítulo 1

Blossom algorithm

The main algorithm[1] to solve the **maximum matching** problem is the algorithm **Blossom** developed by Jack Edmonds in 1965. Hence, this chapter intends to: (1) introduce graph concepts relevant to solving the maximum matching problem; (2) Prove the Blossom algorithm; (3) Implement the Blossom algorithm and (4) present applications of the Blossom algorithm.

1.1 Matchings

1.1.1 Graph: A **graph** G is a triple (V, E, φ) such that

(i) V is the **vertex set**;

(ii) E is the **edge set**;

(iii) $\varphi : E \to V \times V$ is a relation between each edge and a pair of vertices, called the **incidence function** of G.

Usually, it is used $V(G)$ or V_G to denote V and $E(G)$ or E_G to denote E. Also, if $e \in E(G)$ and $\varphi(e) = (u, v)$, then u and v are the **ends** of e; When the context is clear, (u, v) may be abbreviated to uv.

1.1.2 Walk: For a graph $G := (V, E, \varphi)$, a **walk** is a sequence

$$\langle v_0, e_1, v_1, \dots, a_l, v_l \rangle =: W$$

such that

(i) $l \in \mathbb{N}$ is the *length* of W;

(ii) $v_0, v_1, \dots, v_l \in V$;

(iii) $e_1, \dots, e_l \in E$.

[1] If the graph is **guaranteed** to be bipartite, one may search *Kuhn's algorithm* and *Flow Networks*.

It is denoted that $V(W) := \{v_0, \dots, v_l\}$ and $E(W) := \{e_1, \dots, e_l\}$. It is said that W is walk from v_0 to v_l or a (v_0, v_l)-walk. The walk W is a:

- **path**, if all its vertices are distinct;
- **cycle**, if $v_0 = v_l$ and it is an **odd cycle** if its length is odd, else it is an **even cycle**.

1.1.3 Bipartite graphs: A graph G is **bipartite** if there are two sets $U, W \in V(G)$ such that

(i) $U \cap W = \emptyset$;

(ii) $U \cup W = V(G)$;

(iii) every edge of G has one end at U and the other end at W.

In this case, it is said that G is (U, W)-bipartite.

Theorem 1.1.4 (Characterization of bipartite graphs): *A graph is bipartite if and only if it has no odd cycles.*

Proof. □

1.1.5 Matching: For a graph $G := (V, E, \varphi)$, a set $M \subsetneq E$ is a **matching** of G if and only if no two edges in M share an end. A vertex $v \in V$ is M-covered if some edge of M incides in v, and it is said that M covers v; Otherwise, v is M-exposed. The matching M is:

- **maximal**, if there is no edge $e \in E \setminus M$ such that $M \cup \{e\}$ is a matching of G;
- **maximum**, if for every matching M' of G one has $|M| \geq |M'|$;
- **perfect**, if $2|V_G| = |M|$, i.e., every vertex of G is covered.

Denote $\nu(G)$ as the size of a maximum matching in G.

1.2 Maximum matching problem

Now, the maximum matching problem can be described as:

MaxMatching
Given a graph $G := (V, E, \varphi)$ find a maximum matching of G.

1.2.6 Alternating and augmenting paths: Given a matching M of a graph G. A path P is M-**alternating** if its edges are alternating in and out of M. Formally,

$$e_i \in M \iff e_{i+1} \notin M \text{ for each } i \in [l-1]^{2}$$

And P is M-**augmenting** if both v_0 and v_l are M-exposed.

[2] For $n \in \mathbb{N}$, we denote the set $\{1, \dots, n\}$ as $[n]$.

Theorem 1.2.7 (Berge's theorem): *Let $G := (V, E, \varphi)$ be a graph. A matching M is maximum if and only if there are no M-augmenting path.*

Proof. □

1.2.8 Vertex cover: For a graph $G := (V, E, \varphi)$, a subset $K \subseteq V(G)$ is a **vertex cover** of G if every edge of $E(G)$ has an end in in K. A vertex cover is said to be **minimal** if one has $|K| \leq |K'|$ for every vertex cover K' of G. Denote $\tau(G)$ as the size of a minimum vertex cover of G.

Corollary 1.2.9 (Maximum matching upperbound): *Let G be a graph, M be a matching of G and K be a vertex cover of G. Then, $|M| \leq |K|$.*

Proof. □

Theorem 1.2.10 (König's matching theorem): *Let G be a bipartite graph, then the maximum size of a matching of G is equal to the minimum size of a vertex cover of G.*

Proof. □

Note that König's theorem **does not** hold for all graphs; It suffices to consider a single odd cycle.

1.2.1 Odd cycles

Índice remissivo